UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/082,794	02/22/2002	David Bau III	41016.P008	2046

25943 7590 06/06/2006

SCHWABE, WILLIAMSON & WYATT, P.C.
PACWEST CENTER, SUITE 1900
1211 SW FIFTH AVENUE
PORTLAND, OR 97204

EXAMINER	
RUTTEN, JAMES D	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 06/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED
JUN 0 6 2006
Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/082,794
Filing Date: February 22, 2002
Appellant(s): BAU ET AL.

Robert C. Peck
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 13 March 2006 appealing from the Office action mailed 10 August 2005.

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments

The appellant's statement of the status of amendments after final rejection contained in the brief is correct. To further clarify Appellant's statements, the Examiner submits that Appellant's Response to Final Action, filed 05 October 2005, was fully considered, but did not place the application in condition for allowance, which resulted in the mailing of the Advisory Action, mailed 26 October 2005.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

U.S. Patent Application 10/082,807	Bau et al.	February 22, 2002
U.S. Patent Application 10/784,492	Marvin et al.	February 23, 2004
"Using WebLogic Enterprise JavaBeans"	BEA Systems	September 1999
"EJBDoclet"	dreamBean Software	December 21, 2000
"Enterprise JavaBeans"	Monson-Haefel	March 2000
U.S. Patent 5,812,768	Pagé et al.	September 22, 1998
U.S. Patent 6,230,160	Chan et al.	May 8, 2001
Background of the Invention (Pages 1-3 of Appellant's originally filed specification)	Bau et al.	February 22, 2002

(9) Grounds of Rejection

A complete copy of the text of the applied rejections from the Final Rejection mailed 10 August 2005 is included below in item (12).

Claims 1-52 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1, 3-22, 24-30, 33, 35-38, 41-63, 66-75, 77-80, 83, and 84 of copending Application No. 10/082,807 (hereinafter "the '807 application").

Claims 1-4, 10-12, 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39, 41, 44-46 and 48, are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1-8, 19-23, 26, 27, 31-36, 38, 39, 43, and 44 of copending Application No. 10/784,492 (hereinafter "the '492 application").

Claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 are rejected under 35 U.S.C. 102(b) as being anticipated by prior art of record "Using WebLogic Enterprise JavaBeans" by BEA Systems (hereinafter *BEA WebLogic*).

Claims 2 and 3 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of "EJBDoclet", December 21 2000, by dreamBean Software (hereinafter "dreamBean").

Claims 5-8, 18, 23-25, 28-30, 40, 45-47, and 50-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of "Enterprise JavaBeans" by Monson-Haefel (hereinafter "Monson-Haefel").

Claim 9, 19 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of prior art of record U.S. Patent 5,812,768 to Pagé et al. (hereinafter "Page").

Claims 12, 31, and 34 rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of U.S. Patent 6,230,160 to Chan et al. (hereinafter "Chan").

Claims 13, 20, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claim 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of the "Background of the Invention" section appearing on pages 1-3 of the originally filed specification (hereinafter "BOTI").

Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic and BOTI as applied to claims 13, 20, and 42 above, and further in view of Page.

Claims 15, 21, 26, 27, 43, 48 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic and BOTI as applied to claims 13, 20, and 42 above, and further in view of Monson-Haefel.

Claims 32 and 33 rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic and Chan as applied to claims 12, 31, and 34 above, and further in view of dreamBean.

Claim 35 is rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic and Chan as applied to claim 12, 31, and 34 above, and further in view of BOTI.

Claim 36 is rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic, Chan and BOTI as applied to claim 35 above, and further in view of Page.

Claim 37 is rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic, Chan, and BOTI as applied to claim 36 above, and further in view of Monson-Haefel.

(10) Response to Argument

The Appellant's arguments, appearing on pages 5-17 of the Appeal Brief filed 03/13/2006, appear largely similar to arguments presented in the amendment filed 05/13/2005, as well as the after-final response filed 10/05/2005. These arguments have been addressed in the Final Action mailed 08/10/2005, and the Advisory Action mailed 10/26/2005.

As such, it appears that Appellant has chosen not to rebut the examiner's positions in either the Final or Advisory Actions. Thus, without an indication of reasons pointing out any supposed errors in the examiner's position, the current response is essentially reproduced from these earlier actions. Appellant's Appeal Brief (pp. 5-17) was presented with arguments under Roman numeral headings I-XIV, which will be addressed below.

I. <u>Provisional rejection of claims 1-52 under the judicially created doctrine of obviousness-type double patenting was improper because claims 1-52 are patentably distinct from the claims of '807.</u>

This argument was first presented on pages 15-16 in the 5/13/05 filing, and addressed in paragraph 6 on page 3 in the 8/10/05 Final Action. Appellant essentially argues that the provisional double patenting rejection over copending Application No. 10/082,807 (hereinafter '807) should be withdrawn since the '807 application is drawn toward *asynchronous* web services, while the instant application is drawn toward *stateful* web services. This argument is not convincing, since claim 4 of the '807 application explicitly claims "...a stateful conversation between the client and the web service..." Appellant has not pointed out why the "stateful conversation" of the '807 application does not meet the language of the instant application. Rather, Appellant provides the following statement (Brief page 6, 1st paragraph):

> More specifically, exemplary distinctions include that "stateful" web services would include a series of related web service requests, while "asynchronous" web services require coordination as they do not return immediate results.

It should be noted that such language is not found in the claims.

II. Provisional rejection of claims 1-4, 10-12, 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39, 41, 44-46, and 48 under the judicially created doctrine of obviousness-type double patenting was improper because claims 1-4, 10-12, 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39, 41, 44-46 and 48 are patentably distinct from the claims of '492.

This argument was first presented on pages 16 and 17 of the 5/13/05 filing, and addressed in paragraph 7 on page 3 in the 8/10/95 Final Action. Appellant essentially argues that the provisional double patenting rejection over copending Application No. 10/784,492 (hereinafter '492) should be withdrawn since the instant application merely dominates the '492 application, and is patentably distinct from it. However, even if domination exists, its presence does not preclude double patenting (MPEP 804(II)). In the present case, the '492 application claims each of the elements of the instant application as detailed on pages 5-6 of the 2/9/05 Non-Final Action (reproduced under "Double Patenting" in section 10 below). Thus the argument is not convincing.

III. <u>Rejection of claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 under 35 U.S.C. §102(b) was improper because BEA WebLogic fails to anticipate the claimed invention as claimed in claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44.</u>

A form of this argument was first presented on pages 18 and 19 of the 5/13/05 filing, with an additional argument presented at the top of page 18 of the 10/05/05 filing. These arguments were addressed in paragraphs 8-11 appearing on pages 4-5 of the 8/10/05 Final Action, and further in the 10/26/05 Advisory Action.

In response to Appellant's argument that BEA Weblogic fails to show certain features of the invention, it is noted that the features upon which Appellant relies (i.e., arguendo, as opposed to BEA Weblogic, a developer does not perform all steps – page 10 paragraph 2) are not recited in the rejected claim. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Appellant essentially suggests (paragraph 2 page 10) that due to the reliance on a compiler to generate persistent components, "the generating of at least some of the persistent components can not be performed by a developer". However, the previous Office Actions have addressed this feature in citing the BEA WebLogic reference on page 6 along with "Step 3": "WebLogic EJB compiler". This citation shows the reliance of BEA WebLogic upon a developer to type commands in order to generate components. Thus, this argument is not convincing.

Appellant argues in paragraph 2 page 10 that the business logic of BEA WebLogic "is encapsulated 'inside a component framework' not 'exposed as part of the stateful web service'".

However, BEA WebLogic teaches that methods are identified for use by developers. See page 3, paragraph 4: "specify how the component should be used -- which users have access to which methods". These methods are *exposed* to users to allow access. Without exposing these methods, users would not be able to use them. Thus, this argument is not convincing.

In paragraph 4 on page 10, Appellant argues that BEA WebLogic "does not teach or suggest using an enhanced 'compiler to generate' automatically one or more Enterprise JavaBeans™ as well as associated deployment descriptors to store and manage such conversational states based at least in part on 'one or more declarative annotations' as recited in claim 1 of the instant application." Here, Appellant's arguments appear to be directed towards pointing out that the invention described in the originally filed specification allows a developer to supply a relatively small amount of information in the form of source code, and an enhanced compiler then supplies the rest of the source code automatically. However, these features are not recited in the rejected claims. Claim 1 merely calls for:

> **providing** a source code representation of **at least a portion** of web service logic, the logic including one or more methods;
> ...
> **specifying** one or more declarative annotations **to cause a compiler to generate** one or more persistent components ...
> [*Emphasis Added*]

Although this could be interpreted as one or two lines, methods, classes, etc., this could also be interpreted as the full source code. In regard to the "enhanced compiler", claim 1 merely calls for "a compiler to generate one or more persistent components...". Thus, it is clear that the plain language of them claims does not include any reading as to "automatically", nor does it exclude any manual means of code generation. Further, the plain language of the claim does not include any reading for a compiler that generates *source* code. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Appellant's arguments regarding claims 1, 16, and 38 in paragraph 1 and 2 on page 11 fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. The rejection of claims 1, 16 and 38 are provided below for an interpretation of the limitations as found in BEA WebLogic.

IV. <u>Rejection of claims 2, 3, 5-9, 12-15, 18-21, 23-37, 40-43, and 45-52 under 35 U.S.C. §103(a) was improper because the cited references, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 2, 3, 5-9, 12-15, 18-21, 23-37, 40-43, and 45-52 is viewed as a whole.</u>

These arguments are based upon the Appellant's arguments regarding the BEA WebLogic reference in connection with claim 1 presented above. Likewise, these arguments are not persuasive for the reasons set forth above.

(11) Related Proceeding(s) Appendix

To the best of the examiner's knowledge, there are no related appeals or interference proceedings currently pending, which would directly affect or be directed affected by or have a bearing on the Board's decision in this appeal.

(12) Text of Final Rejection

The text of the Final Action mailed 8/10/2005, is reproduced below for completeness.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1-52 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1, 3-22, 24-30, 33, 35-38, 41-63, 66-75, 77-80, 83, and 84 of copending Application No. 10/082,807 (hereinafter "the '807 application"). Although the conflicting claims are not identical, they are not patentably distinct from each other because with respect to claim 1, for example, the '807 application discloses:

1. A method of specifying a stateful web service within a procedural programming environment, the method comprising:

providing a source code representation of at least a portion of web service logic, the logic including one or more methods; See page 32 lines 5 and 6:

> providing a source code representation of at least a portion of web service logic, the logic including at least one method

identifying one of said one or more methods to be exposed as part of the stateful web service; See page 32 line 7:

> identifying a member variable declared to implement said callback method

Also see page 32 line 18: "conversational state"

specifying one or more declarative annotations to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified method. See page 32 lines 16-18:

> specifying one or more declarative annotations associated with said callback method to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified member variable.

The '807 application does not expressly disclose "identifying one of said one or more methods". However, it does teach "identifying a member variable." In this context, the "member variable" is implementing a method. As such, this "member variable" can be interpreted as a method. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the '807 application's "member variable" with the present application's methods. One of ordinary skill would have been motivated to provide support for callback methods as well as any other type of method.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claims 1-4, 10-12, 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39, 41, 44-46 and 48, are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1-8, 19-23, 26, 27, 31-36, 38, 39, 43, and 44 of copending Application No. 10/784,492 (hereinafter "the '492 application"). Although the conflicting claims are not identical, they are not patentably distinct from each other because, for example, the '492 application discloses:

1. A method of specifying a stateful web service within a procedural programming environment (see page 39 lines 4-12)*, the method comprising:*

providing a source code representation of at least a portion of web service logic, the logic including one or more methods; See page 36 lines 2-4:

> an annotated source code, which is a programming language augmented with declarative meta-data capable of exposing program logic as a network-accessible service

identifying one of said one or more methods to be exposed as part of the stateful web service; See page 36 lines 5-6:

> at least one deployed service component capable of providing the network-accessible service to a client

Further, see page 37 lines 5-6:

> the annotated source code is capable of facilitating access to an external service, which can be one of stateful, stateless, synchronous, and asynchronous.

specifying one or more declarative annotations to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified method. See page 36 lines 7-10:

> an enhanced compiler capable of analyzing the annotated source code, recognizing numerous types of **meta-data annotations**, and generating a mechanism, which can include one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service **component**

Further, see page 36 lines 14-16:

> the system is capable of simultaneously managing multiple transactions, wherein each transaction can be a **conversation** of a request and/or a response from the client for the network-accessible service.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 102

Claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 are rejected under 35 U.S.C. 102(b) as being anticipated by prior art of record "Using WebLogic Enterprise JavaBeans" by BEA Systems (hereinafter *BEA WebLogic*).

In regard to claim 1, BEA WebLogic discloses:

A method of specifying a stateful web service within a procedural programming environment, (See page 5 steps 1-3) *the method comprising:*

providing a source code representation of at least a portion of web service logic, the logic including one or more methods; See section III on page 4:

> There are three parts to using WebLogic EJB:
> 1. Develop an EJBean or obtain one from a third-party supplier.

Also see page 3 paragraph 5 for disclosure of methods in an EJB:

> ...an EJBean contains the business logic (**methods**)...

identifying one of said one or more methods to be exposed as part of the stateful web service; See page 2:

> Session beans (either **stateful** or stateless)

Also see page 3, 4th paragraph:

> With the EJB model, you can write or buy business components (such as invoices, bank accounts and shipping routes) and, during **deployment** into a certain project, specify how the component should be used -- which users have **access to which methods**, whether the framework should automatically start a transaction or whether it should inherit the caller's transaction, and so on.

Also page 6 "Step 2" discloses identification of methods to be exposed:

> Check the deployment descriptor and modify any of its properties for your particular deployment (if required).

specifying one or more declarative annotations to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified method.

See page 5 "Step 2":

> The **Deployment Descriptor** ties together the different classes and interfaces, and is used to **build the code-generated class files**. It also allows you to specify some aspects of the EJBean's **deployment** at runtime.

Also see page 6 along with "Step 3":

> Generate the wrapper classes using the WebLogic EJB **compiler** (ejbc)...
> This will **create the appropriate files** for the bean...

Also, page 8 discloses the general capabilities of EJBeans with respect to persistence and transactional, or "conversational", state:

> An entity EJBean can save its state in any **transactional** or non-transactional **persistent** storage...

In regard to claim 4, the above rejection of claim 1 is incorporated. BEA WebLogic further discloses: *wherein the one or more declarative annotations are specified outside of the source code representation and associated with the identified method by the compiler.* See page 6 "Step 3".

In regard to claim 10, the above rejection of claim 1 is incorporated. BEA WebLogic further discloses: *wherein the one or more declarative annotations are manually specified by a developer.* See page 6 "Step 2".

In regard to claim 11, the above rejection of claim 1 is incorporated. BEA WebLogic further discloses: *wherein the one or more declarative annotations are automatically specified by an integrated development environment based upon input provided by a developer.* See page 5 "Step 2".

In regard to claim 16, BEA WebLogic discloses:

In a procedural programming environment, a method of generating a stateful web service (See pages 6 and 7)*, the method comprising:*

reading on one or more computing devices a segment of procedural source code representing at least a portion of the web service; parsing on one or more computing devices the segment of source code to identify the presence of one or more declarative annotations identifying an associated method within the segment as being stateful; generating on one or more computing devices one or *more object codes defining one or more publicly accessible service components based at least in part upon the source code;* See page 6 "Step 3":

> **Generate the wrapper classes** using the WebLogic EJB compiler (ejbc) with this command (typed on one line), **referencing the serialized deployment descriptor**:
>
> ```
> $ java weblogic.ejbc -d /weblogic/myserver/temp AccountBeanDD.ser
> ```
>
> ...
>
> This will create the appropriate files for the bean, and place them in a temporary directory

Reading and parsing source code is an inherent feature of a compiler, as object code could not be generated without both steps. This passage also shows use of a computing device by the invocation of a command that is "typed on one line".

Generating on one or more computing devices meta-data based at least in part upon the one or more declarative annotations; associating on one or more computing devices meta-data with the one or more object codes. See page 5 "Step 2":

> The Deployment Descriptor **ties together** the different classes and interfaces, and is **used to build** the code-generated class files.

In regard to claim 17, the above rejection of claim 16 is incorporated. All further limitations have been addressed in the above rejection of claim 1.

In regard to claim 22, the above rejection of claim 16 is incorporated. BEA WebLogic further discloses: *wherein the source code is written in the Java programming language.* BEA WebLogic discloses implementation using Enterprise JavaBeans (EJB - See page 2) which is an API that uses the Java programming language.

In regard to claim 38, BEA WebLogic discloses:

An article of manufacture comprising: a storage medium having stored therein a plurality of programming instructions. Page 7 step 5 shows a directory path for storage of programming instructions as cited in the rejection of claim 31. All further limitations have been addressed in the above rejection of claim 16.

In regard to claim 39, the above rejection of claim 38 is incorporated. All further limitations have been addressed in the above rejection of claim 17.

In regard to claim 44, the above rejection of claim 38 is incorporated. All further limitations have been addressed in the above rejection of claim 22.

Claim Rejections - 35 USC § 103

Claims 2 and 3 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of "EJBDoclet", December 21 2000, by dreamBean Software (hereinafter "dreamBean").

In regard to claim 2, the above rejection of claim 1 is incorporated. BEA WebLogic does not expressly disclose: *wherein the one or more declarative annotations are specified within the source code representation.* However, in an analogous environment, dreamBean teaches a declarative annotation within the source code. See page 2 under the heading "Custom EJBDoclet tags". It would have been obvious to one of ordinary skill in the art at the time the invention was made to use dreamBean's declarative annotations within BEA WebLogic's source code. One of ordinary skill would have been motivated to automatically generate a remote interface (see dreamBean page 1 under "Features").

In regard to claim 3, the above rejection of claim 2 is incorporated. BEA WebLogic does not expressly disclose declarative annotations within a comment field preceding an identified method. However, in an analogous environment, dreamBean teaches a tool for generating "EJB files from a commented bean source-file" (page 1 paragraph 1). dreamBean further teaches using comment fields preceding a method to support generation of externally accessible methods. See middle of page 5, where the annotation "@remote-method" appears in a comment preceding the identified methods "deposit" and "withdraw". It would have been obvious to one of ordinary skill in the art at the time the invention was made to use dreamBean's declarative annotation with BEA WebLogic's source code. One of ordinary skill would have been motivated to automatically generate a remote interface (see dreamBean page 1 under "Features").

Claims 5-8, 18, 23-25, 28-30, 40, 45-47, and 50-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of "Enterprise JavaBeans" by Monson-Haefel (hereinafter "Monson-Haefel").

In regard to claim 5, the above rejection of claim 1 is incorporated. BEA WebLogic further discloses use of the "ejbCreate" function. See page 24, 4th paragraph. BEA WebLogic does not expressly disclose specifics of the start, continue or finish methods. However, in an analogous environment, Monson-Haefel teaches that "deployment descriptors" could be used as declarative annotations that indicate: *wherein the start method applies to the start of a stateful conversation between a client and the web service* (see section 7.4.2.1), *the continue method applies to the continuation of an ongoing stateful conversation between a client and the web service* (see section 7.4.2 "ejbActivate()" on page 4 of section 7.4), *and the finish method applies to the completion of an ongoing stateful conversation between a client and the web service* (see section 7.4.2.3). Monson-Haefel further teaches that such methods can be entered as annotations in section 10.6.3.2. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Monson-Haefel's teaching of stateful sessions with BAE Websphere's components. One of ordinary skill would have been motivated to provide a dedicated stateful session bean to act on behalf of a client for its entire life cycle (see section 7.3 paragraph 1).

In regard to claim 6, the above rejection of claim 5 is incorporated. BEA WebLogic does not expressly disclose: *wherein when a method declared to be a start method is invoked at run-time, a new instance of a conversation is created, and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations*. However, Monson-Haefel teaches instantiation of a conversation and return of an identifier upon invocation of a start method. See Section 7.4.2.1.

In regard to claim 7, the above rejection of claim 5 is incorporated. BEA WebLogic does not expressly disclose: *wherein when a method declared to be a continue method or a finish method is invoked at run-time, a unique identifier provided by the client is obtained and used to access a corresponding instance of a conversation*. However, Monson-Haefel teaches the return of an identifier as noted in the above rejection of claim 6. Use of the representative identifier is inherent in referencing the session as discussed in section 7.4.2.3.

In regard to claim 8, the above rejection of claim 7 is incorporated. BEA WebLogic does not expressly disclose: *wherein when a finish method is invoked at run-time, the corresponding instance of the conversation is destroyed after processing by the web service logic*. However, Monson-Haefel teaches that the instance is destroyed after processing. See section 7.4.2.3.

In regard to claim 18, the above rejection of claim 16 is incorporated. All further limitations have been addressed in the above rejection of claim 5.

In regard to claim 23, BEA WebLogic discloses:

In a stateful web service, a method (See pages 6 and 7) *comprising:*

receiving a message requesting that a web service method be invoked; See page 14 3rd paragraph:

> When clients lookup and obtain a reference to an EJBean, new instances are created, but with only one instance per primary key.

In order for a reference to the EJBean to be returned, a message must have been received.

parsing the message to identify the requested method; See page 14, 3rd paragraph as cited above. Parsing is inherent in receiving a message, otherwise it could not be understood.

dispatching the received request to invoke the identified stateful method. See page 14 3rd paragraph as cited above.

BEA WebLogic further discloses the use of deployment descriptors. See page 20 "DDCreator: Deployment descriptor generator". BEA WebLogic does not expressly disclose: *determining whether the method is a stateful method based at least in part upon meta-data derived from one or more declarative annotations stored in association with object codes of the web service.* However, Monson-Haefel teaches that deployment descriptors can be used to identify a stateful method. See section 7.3.1.7:

> The most important difference between this descriptor and the deployment descriptor used for the ProcessPayment bean is the <session-type> tag, which states that this bean is stateful.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Monson-Haefel's "session-type" tag with BEA WebLogic's deployment descriptor. One of ordinary skill would have been motivated to determine at runtime the type of method being invoked in order to associate the proper protocols and methods with the particular bean, allowing for greater reliability and efficiency.

In regard to claim 24, the above rejection of claim 23 is incorporated. BEA WebLogic further discloses: *wherein the message is received from a remote client*. See page 14 3rd paragraph as cited above.

In regard to claim 25, the above rejection of claim 23 is incorporated. All further limitations have been addressed in the above rejection of claim 6.

In regard to claims 28-30, the above rejection of claim 23 is incorporated. All further limitations have been addressed in the above rejections of claims 6-8, respectively.

In regard to claim 40, the above rejection of claim 38 is incorporated. All further limitations have been addressed in the above rejection of claim 18.

In regard to claim 45, all limitations have been addressed in the above rejections of claims 23 and 31.

In regard to claims 46 and 47, the above rejection of claim 45 is incorporated. All further limitations have been addressed in the above rejections of claims 24 and 25, respectively.

In regard to claims 50-52, the above rejection of claim 45 is incorporated. All further limitations have been addressed in the above rejections of claims 28-30, respectively.

Claim 9, 19 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of prior art of record U.S. Patent 5,812,768 to Pagé et al. (hereinafter "Page").

In regard to claim 9, the above rejection of claim 1 is incorporated. BEA WebLogic does not expressly disclose: wherein the one or more declarative annotations indicate to the compiler whether the identified method is buffered, wherein if the identified method is buffered the compiler instantiates one or more queues to temporarily store one or more requests for the identified method. However, in an analogous environment, Page teaches that interaction with web services can be implemented as buffered messages that operate asynchronously via message queues. See column 6 lines 39-46. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Page's queues with BEA WebLogic's methods. One of ordinary skill would have been motivated to allow "store and forward" technology that would enable greater scalability.

In regard to claim 19, the above rejection of claim 16 is incorporated. All further limitations have been addressed in the above rejection of claim 9.

In regard to claim 41, the above rejection of claim 38 is incorporated. All further limitations have been addressed in the above rejection of claim 19.

Claims 12, 31, and 34 rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of U.S. Patent 6,230,160 to Chan et al. (hereinafter "Chan").

In regard to claim 12, the above rejection of claim 11 is incorporated. BEA WebLogic does not expressly disclose: *wherein said input includes graphical manipulation of the identified method by the developer via the integrated development environment.* However, in an analogous environment, Chan teaches an IDE for manipulation of program elements and methods. See FIG. 4A and column 8 lines 19-30.

In regard to claim 31, BEA WebLogic discloses:

a storage medium having stored therein a plurality of programming instructions (page 7 step 5 shows a directory path for storage of programming instructions), BEA WebLogic does not expressly disclose: *which when executed provide a graphical interface to facilitate specification*. All further limitations have been addressed in the above rejection of claim 1.

However, Chan teaches that a graphical user interface can be used to facilitate specification. See FIG. 4A and column 8 lines 19-21:

> FIG. 4A is a window view from a **visual building tool** listing the methods, properties and events programmed for a server bean of the type discussed herein.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Chan's teaching of a visual building tool with BEA WebLogic's annotated code. One of ordinary skill would have been motivated to provide an intuitive interface to simplify and speed-up code development.

In regard to claim 34, the above rejection of claim 31 is incorporated. All further limitations have been addressed in the above rejection of claim 4.

Claims 13, 20, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic as applied to claim 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 above, and further in view of the "Background of the Invention" section appearing on pages 1-3 of the originally filed specification (hereinafter "BOTI").

In regard to claim 13, the above rejection of claim 1 is incorporated. BEA WebLogic does not expressly discloses: *a proxy object designed to facilitate interaction by the web service with one of an external web service or client.* However, BOTI teaches implementation of proxy objects. See page 2 lines 17-19. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of implementation of a proxy object with BEA WebLogic's web service. One of ordinary skill would have been motivated to automatically generate a required proxy mechanism.

In regard to claim 20, the above rejection of claim 16 is incorporated. All further limitations have been addressed in the above rejection of claim 13.

In regard to claim 42, the above rejection of claim 38 is incorporated. All further limitations have been addressed in the above rejection of claim 20.

Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic and BOTI as applied to claims 13, 20, and 42 above, and further in view of Page.

In regard to claim 14, the above rejection of claim 13 is incorporated. All further limitations have been addressed in the above rejection of claim 9.

Claims 15, 21, 26, 27, 43, 48 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic and BOTI as applied to claims 13, 20, and 42 above, and further in view of Monson-Haefel.

In regard to claims 15 and 21, the above rejection of claims 13 and 20 are respectively incorporated. All further limitations have been addressed in the above rejection of claim 6.

In regard to claim 26, the above rejection of claim 23 is incorporated. BEA WebLogic does not expressly disclose: *wherein the message is a SOAP based message*. However BOTI teaches SOAP based messages on page 2 lines 8 and 9.

In regard to claim 27, the above rejection of claim 26 is incorporated. BEA WebLogic does not expressly disclose: wherein the conversational identifier is a GUID encapsulated in a header of the SOAP message. BOTI teaches encapsulation using SOAP as addressed in the above rejection of claim 26, and Monson-Haefel teaches a GUID. It would have been obvious to one of ordinary skill in the art at the time the invention was made to encapsulate Monson-Haefel's identifier in BOTI's SOAP message. One of ordinary skill would have been motivated to use SOAP's encapsulation as a way to transfer instance identifier information using a platform independent representation.

In regard to claim 43, the above rejection of claim 42 is incorporated. All further limitations have been addressed in the above rejection of claim 21.

In regard to claims 48 and 49, the above rejection of claim 45 is incorporated. All further limitations have been addressed in the above rejections of claims 26 and 27, respectively.

Claims 32 and 33 rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic and Chan as applied to claims 12, 31, and 34 above, and further in view of dreamBean.

In regard to claim 32, the above rejection of claim 31 is incorporated. All further limitations have been addressed in the above rejection of claim 2 and 12.

In regard to claim 33, the above rejection of claim 32 is incorporated. All further limitations have been addressed in the above rejection of claim 3.

Claim 35 is rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic and Chan as applied to claim 12, 31, and 34 above, and further in view of BOTI.

In regard to claim 35, the above rejection of claim 31 is incorporated. All further limitations have been addressed in the above rejection of claim 13.

Claim 36 is rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic, Chan and BOTI as applied to claim 35 above, and further in view of Page.

In regard to claim 36, the above rejection of claim 35 is incorporated. All further limitations have been addressed in the above rejection of claim 14.

Claim 37 is rejected under 35 U.S.C. 103(a) as being unpatentable over BEA WebLogic, Chan, and BOTI as applied to claim 36 above, and further in view of Monson-Haefel.

In regard to claim 37, the above rejection of claim 35 is incorporated. All further limitations have been addressed in the above rejection of claim 15.

--End Text of Final Rejection--

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

J. Derek Rutten

Conferees:

[illegible]

TUAN DAM
SUPERVISORY PATENT EXAMINER

Tuan Q. Dam

[illegible]

Wei Y. Zhen

WEI ZHEN
SUPERVISORY PATENT EXAMINER